MATHEMATICAL METHODS
IN PATTERN RECOGNITION

Accurate Estimates of the Generalizing Ability for Symmetric Sets of Algorithms and Randomized Learning Methods

A. I. Frei

Moscow Institute of Physics and Technology, Institutskii per. 9, Dolgoprudny, Moskovskaya oblast', 141700 Russia
e-mail: sfrey@yandex.ru

Abstract—The main issue of the combinatorial approach to overtraining is to obtain computationally efficient formulas for overtraining probabilities. A group-theoretical approach is proposed to simplify derivation of such formulas when the set of algorithms has a certain group of symmetries. Examples of the sets are given. The general estimate of overtraining probability is proved for the randomized learning method. It is applied to four model sets of algorithms—a layer of the Boolean cube, the Boolean cube, the unimodal chain, and a bundle of monotonic chains.

Keywords—statistical learning theory, generalizing ability, overtraining probability, empirical risk minimization, randomized learning method, group of symmetries, group orbit, monotonic chain of algorithms, bundle of monotonic chains.

DOI: 10.1134/S1054661810030016

1. INTRODUCTION

The problem of choice under incomplete information always accompanies pattern recognition, regression restoration, and forecasting problems. Given only a finite learning sample of objects, one needs to choose the algorithm from the given set of algorithms that would make as few errors as possible on the objects from both the observable learning sample and the hidden control sample not known when the algorithm is chosen. If the frequency of errors of the control sample exceeds that of the learning sample, we speak of overtraining or overfitting of the algorithm, i.e., being too good in describing particular data, it cannot generalize it, restore the pattern that generated it, or be used to make predictions.

The frequency of errors on the learning sample is also called *empirical risk*. *Empirical risk minimization* is the learning method that chooses such algorithm from the given set that makes the fewest number of errors on the learning sample [1, 2]. The table gives an example of empirical risk minimization leading to overtraining. The columns of the table stand for the algorithms; its rows, for the objects of the learning sample $\{x_1, x_2, x_3\}$ and the control sample $\{x_4, x_5, x_6\}$. The unity in the $[i, d]$-th cell of the table means that the algorithm a_d makes an error on the object x_i.

$$\begin{array}{c|cccccc|}
 & a_1 & a_2 & \dots & a_d & \dots & a_d \\
x_1 & 0 & 1 & \dots & 0 & \dots & 1 \\
x_2 & 1 & 1 & \dots & 0 & \dots & 0 \\
x_3 & 0 & 0 & \dots & 0 & \dots & 0 \\
\hline
x_4 & 1 & 1 & \dots & 1 & \dots & 1 \\
x_5 & 1 & 0 & \dots & 1 & \dots & 0 \\
x_6 & 0 & 0 & \dots & 1 & \dots & 0
\end{array}.$$

In this example, overtraining may result from a "bad" decomposition of the general sample into learning and control samples. Therefore, we introduce the functional of *overtraining probability* that equals the part of decompositions of the sample that cause overtraining [3, 4]. This functional is invariant to the choice of the decomposition and characterizes the quality of the learning method for the given general sample.

For some families of simple structure (monotonic and unimodal chains and h-dimensional grids), accurate expressions of the overtraining probability are found in [3, 5]. In this work, we develop the group-theoretical approach [6] that helps estimate the overtraining probability efficiently for sets of algorithms possessing symmetry properties.

1.1. Definitions

Let the general sample $\mathbb{X} = (x_1, ..., x_L)$ consisting of L objects be given. An arbitrary classification algorithm applied to this sample generates a binary vector

Received April 14, 2010

ISSN 1054-6618, Pattern Recognition and Image Analysis, 2010, Vol. 20, No. 3, pp. 241–250. © Pleiades Publishing, Ltd., 2010.

of errors $a \equiv (a(x_i))_{i=1}^{L}$, where $a(x_i) = 1$ means that the algorithm a makes an error on the object x_i. We assume the general sample $\mathbb{X}$ to be fixed; therefore, the algorithms are identified with their vectors of errors.

We use $\mathbb{A} = \{0, 1\}^L$ to denote the set of all possible vectors of errors of length L. Then, $2^{\mathbb{A}}$ is the set of all subsets of $\mathbb{A}$. Note that $|\mathbb{A}| = 2^L$, $|2^{\mathbb{A}}| = 2^{2^L}$.

We use $[\mathbb{X}]^l$ to denote the set of all decompositions of the general sample $\mathbb{X}$ into the learning sample X of length l and the control sample $\bar{X}$ of length $k = L - l$.

We denote *the number of errors* of the algorithm a on the sample $U \subseteq \mathbb{X}$ by $n(a, U) = \sum_{x \in U} a(x)$.

An arbitrary mapping of the form μ: $2^{\mathbb{A}} \times [\mathbb{X}]^l \longrightarrow \mathbb{A}$ is called *the deterministic learning method.* The learning method μ uses the learning sample X to choose some algorithm $a = \mu(A, X)$ from the subset $A \subseteq \mathbb{A}$. The learning method is called *empirical risk minimization* if the algorithm it returns makes a minimal number of errors in the course of training; i.e., for all $X \in [\mathbb{X}]^l$ and $A \subseteq \mathbb{A}$ $\mu(A, X) \in A(X)$ holds, where

$$A(X) = \operatorname*{Argmin}_{a \in A} n(a, X).$$

Empirical risk minimization can be ambiguous; i.e., several algorithms from $A(X)$ can have the same number of errors on the learning sample. In [4], *pessimistic* empirical risk minimization was used to eliminate ambiguity and obtain accurate upper estimates of the overtraining probability, assuming that the algorithm with the maximal number of errors on the general sample $\mathbb{X}$ is chosen in the case of ambiguity. This does not eliminate ambiguity for good. There can be cases when several algorithms have a minimal number of errors on the learning sample X and the same number of errors on the general sample $\mathbb{X}$. In such cases, the set of algorithms had a linear order introduced for it, with the algorithm with a greater order number chosen among indistinguishable algorithms. Setting priorities for the algorithms is artificial, with no proper analogues among known learning methods.

1.2. Randomized Learning Method

A randomized learning method matches an arbitrary set of algorithms $A \subseteq \mathbb{A}$ and an arbitrary learning sample $X \in [\mathbb{X}]^l$ with the weight distribution function on the set of algorithms

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^l \longrightarrow \{f : \mathbb{A} \longrightarrow [0, 1]\}. \qquad (1)$$

Naturally, we can assume this function to be normalized and interpreted as the probability to obtain each algorithm as a result of training.

The deterministic learning method is a special case of the randomized method when the weight distribution function $f(a)$ takes the unity value exactly on one algorithm and the zero value on all other algorithms.

Note that there is an equivalent way to give the mapping from definition (1)

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^l \times \mathbb{A} \longrightarrow [0, 1].$$

We consider the group S_L, which is a symmetric group consisting of L elements and acting on the set of objects of the general sample by permutations $S_L = \{\pi$: $\mathbb{X} \longrightarrow \mathbb{X}\}$.

For each $\pi \in S_L$, we give the way π acts upon the arbitrary sample $X \in [\mathbb{X}]^l$ as a mapping π: $\mathbb{X} \longrightarrow \mathbb{X}$ that acts upon each object of the sample X: $\pi X = \{\pi x$: $x \in X\}$ on an element-wise basis. The mapping does not change the number of objects $|X| = |\pi X|$, so we can speak about π acting on the set of decompositions of the general sample into training and control samples of the fixed length π: $[\mathbb{X}]^l \longrightarrow [\mathbb{X}]^l$.

We assume that, applied to the set of all algorithms $\mathbb{A}$, S_L permutates vectors of errors of algorithms $(\pi a)(x_i) = a(\pi^{-1}x_i)$. Here, objects are subjected to the reciprocal permutation π^{-1} since it is in this case when it is correct to say that the group S_L *acts* on the set $\mathbb{A}$.

Lemma 1.1. The number of errors of the algorithm a on the subsample $U \subseteq \mathbb{X}$ remains the same after the permutation $\pi \in S_L$ is simultaneously applied to the algorithm and the subsample

$$n(a, U) = n(\pi a, \pi U). \qquad (2)$$

Proof. We write the definition of the number of errors of the algorithm and use the action of the permutation π on the algorithm a defined above

$$n(\pi a, \pi U) = \sum_{x_i \in \pi U} (\pi a)(x_i) = \sum_{x'_i} (\pi a)(\pi x'_i)$$

$$= \sum_{x'_i \in U} a(\pi^{-1}(\pi x'_i)) = \sum_{x'_i \in U} a(x'_i) = n(a, U).$$

The action of the group S_L on the set of all possible algorithms $\mathbb{A}$ is naturally extended to the system of all subsets S_L: $2^{\mathbb{A}} \longrightarrow 2^{\mathbb{A}}$ by the rule $\pi A = \{\pi a$: $a \in A\}$. In what follows, we use the single designation π for the actions described above.

Now, we can give a stricter definition of a randomized learning method.

Definition 1. We call a *randomized learning method* the mapping of the form

$$\mu: 2^{\mathbb{A}} \times [\mathbb{X}]^l \times \mathbb{A} \longrightarrow [0, 1] \qquad (3)$$

that satisfies the following conditions for any $A \in 2^{\mathbb{A}}$, $X \in [\mathbb{X}]^l$, $a, b \in A$ and $\pi \in S_L$

(1) normalization

$$\sum_{a \in A} \mu(A, X, a) = 1, \qquad (4)$$

(2) algorithms with the same frequency of errors in the course of training are indistinguishable,

$$n(a, X) = n(b, X) \longrightarrow \mu(A, X, a) = \mu(A, X, b), \quad (5)$$

(3) the training result is invariant with respect to replacing the set of algorithms A by $\pi(A)$

$$\mu(A, X, a) = \mu(\pi A, \pi X, \pi a). \quad (6)$$

The first condition implies "probabilistic" normalization of weights of the algorithms and ensures the zero "probabilities" to the algorithms that do not belong to the set A. The second condition means that for any decomposition $\mathbb{X} = X \sqcup \bar{X}$, $X \in [\mathbb{X}]^l$, the probability to obtain the algorithm as a result of training depends only on the number of errors of the algorithm being trained. The third condition means that the training result is not affected if the permutation π acts upon the set of objects $[\mathbb{X}]^l$ and on the set of algorithms $\mathbb{A}$ simultaneously.

The following mapping is a constructive example of the randomized learning method, which we call a *randomized method of empirical risk minimization*

$$\mu(A, X, a) = \frac{[a \in A(X)]}{|A(X)|}. \quad (7)$$

Theorem 1. *Mapping* (7) *is a randomized learning method.*

Proof. We can check the first condition explicitly

$$\sum_{a \in A} \mu(A, X, a) = \sum_{a \in A(X)} \frac{1}{|A(X)|} = 1.$$

To prove the second proposition, it is sufficient to note that two algorithms a_1 and a_2 with the same number of errors in the course of training can belong to the set $A(X)$ only simultaneously. Hence, the probability to obtain each of the algorithms as a result of training is either zero or $\frac{1}{|A(X)|}$.

To prove the third condition, it is sufficient to prove that

$$a_0 \in \operatorname*{Argmin}_{a \in A} n(a, X) \Leftrightarrow \pi a_0 \in \operatorname*{Argmin}_{a \in \pi A} n(a, \pi X).$$

Using Lemma 1.1, we make the following chain of equivalent statements

$$\begin{aligned}
&a_0 \in \operatorname*{Argmin}_{a \in A} n(a, X)\\
&\Leftrightarrow \forall a \in A \longrightarrow n(a_0, X) \le n(a, X)\\
&\Leftrightarrow \forall a \in A \longrightarrow n(\pi a_0, \pi X) \le n(\pi a, \pi X)\\
&\Leftrightarrow \forall a' \in \pi A \longrightarrow n(\pi a_0, \pi X) \le n(a', \pi X)\\
&\Leftrightarrow \pi a_0 \in \operatorname*{Argmin}_{a \in A} n(a, \pi X).
\end{aligned}$$

The theorem is proved.

1.3. Overtraining Probability

The variable $\nu(a, U) = n(a, U)/|U|$ is called *the frequency of errors* of the algorithm a on the sample U. We define the *deviation of frequencies* on the decomposition $\mathbb{X} = X \sqcup \bar{X}$ as the difference between the frequencies for the control and training samples: $\delta(a, X) = \nu(a, \bar{X}) - n(a, X)$.

We fix the parameter $\varepsilon \in (0, 1]$. We say that the algorithm *a is overtrained* for the decomposition $X \sqcup \bar{X}$ if $\delta(a, X) \ge \varepsilon$.

We make the main (and the only) probabilistic assumption that all decompositions of the general sample into observable and hidden subsamples are equiprobable [3, 4].

If φ: $[\mathbb{X}]^l \longrightarrow$ {true, false} is a predicate, we call *the probability of the event* $\varphi(X)$ the fraction of decompositions of the sample such that the predicate $\varphi(X)$ is true

$$\mathbf{P}[\varphi(X)] = \frac{1}{C_L^l} \sum_{X \in [\mathbb{X}]^l} [\varphi(X)].$$

Hence, the mathematical expectation of the arbitrary function x: $[\mathbb{X}]^l \longrightarrow \mathbb{R}$ is

$$\mathbf{E}\xi(X) = \frac{1}{C_L^l} \sum_{X \in [\mathbb{X}]^l} \xi(X).$$

We call the variable

$$P_\mu 9(a, A) = \mathbf{E}\mu(A, X, a) \quad (8)$$

the probability to obtain the algorithm $a \in A$ as a result of training.

For an arbitrary $\varepsilon \in (0, 1]$, we give the *contribution* the algorithm $a \in A$ makes into the overtraining probability

$$Q_\mu(\varepsilon, a, A) = \mathbf{E}\mu(A, X, a)[\delta(a, X) \ge \varepsilon]. \quad (9)$$

We define the *overtraining probability* as the sum of contributions over all algorithms

$$\begin{aligned}
Q_\mu(\varepsilon, A) &= \sum_{a \in A} Q_\mu(\varepsilon, a, A)\\
&= \mathbf{E} \sum_{a \in A} \mu(A, X, a)[\delta(a, X) \ge \varepsilon].
\end{aligned} \quad (10)$$

We can simplify this definition for the deterministic learning method μ: $2^{\mathbb{A}} \times [\mathbb{X}]^l \longrightarrow \mathbb{A}$

$$\begin{aligned}
Q_\mu 9(\varepsilon, A) &= \mathbf{E} \sum_{a \in A} [\mu(A, X) = a][\delta(a, X) \ge \varepsilon]\\
&= \mathbf{E}[\delta(\mu(A, X), X) \ge \varepsilon].
\end{aligned}$$

The obtained expression is literally "the fraction of decompositions of the sample into the learning and control samples such that the chosen algorithm $a = \mu(A, X)$ is overtrained."

Definition 2. Empirical risk minimization methods

$$\mu_o X = \arg\min_{a \in A(X)} n(a, \bar{X}),$$

$$\mu_p X = \arg\max_{a \in A(X)} n(a, \bar{X})$$

are called *optimistic* and *pessimistic*, respectively.

Theorem 2. *Let* μ *be the randomized method of empirical risk minimization. For an arbitrary set of algorithms* $A \subseteq \mathbb{A}$ *and each* $\varepsilon \in (0, 1]$, *the inequalities hold*

$$Q_{\mu_o}(\varepsilon, A) \le Q_\mu(\varepsilon, A) \le Q_{\mu_p}(\varepsilon, A). \tag{11}$$

This theorem lets us call the methods μ, μ_p, and μ_o, respectively, the choice of random, worst, and best algorithms among the algorithms best in the course of learning.

Proof. For the sake of brevity, we write the mappings μ_o and μ_p without the argument A. We show that the proposition holds for each decomposition of the sample

$$[\delta(\mu_o(X), X) \ge \varepsilon]$$

$$\le \sum_{a \in A(X)} \frac{1}{|A(X)|}[\delta(a, X) \ge \varepsilon] \le [\delta(\mu_p(X), X) \ge \varepsilon].$$

We introduce the designations

$$F_o \equiv [\delta(\mu_o(X), X) \ge \varepsilon],$$

$$F_p \equiv [\delta(\mu_p(X), X) \ge \varepsilon],$$

$$F \equiv \frac{1}{|A(X)|} \sum_{a \in A(X)} [\delta(a, X) \ge \varepsilon].$$

We consider the inequality $F_o \le F$. Note that F_o can take only two values, 0 and 1, and the value of the expression F is bounded by the segment [0, 1]. Hence, if $F_o = 0$, the inequality holds automatically.

We prove that $F_o = 1$ follows from $F = 1$. We denote $a_o \equiv \mu_o(X)$. By definition of μ, $a_o \in A(X)$ and $\forall a \in A(X)$ $n(a_o, \bar{X}) \le n(a, \bar{X})$. Hence, $\forall a \in A(X)$ $\delta(a, X) \ge \delta(a_o, X) \ge \varepsilon$. Thus, $F = \sum_{a \in A(X)} \frac{1}{|A(X)|} = 1$.

To prove $F \le F_p$, it is sufficient to consider two cases, $F_p = 0$ and $F_p = 1$, and perform similar reasoning.

2. SYMMETRY OF SETS OF ALGORITHMS

With the concepts introduced above, we can define the group of symmetry of the set of algorithms and use it to obtain computationally efficient formulas of the overtraining probability.

2.1. Invariance of the Overtraining Probability to the Action of the Group S_L

The definitions of the functionals $P_\mu(a, A)$, $Q_\mu(\varepsilon, a, A)$, and $Q_\mu(\varepsilon, A)$ relied on the fact that the objects of the general sample $\mathbb{X}$ are arranged. We prove that these functionals are invariant to the change of numeration of the objects in $\mathbb{X}$.

For the sake of brevity of designations, we omit the argument ε in the function $Q_\mu(\varepsilon, a, A)$.

Lemma 2. *The probability* $P_\mu(a, A)$ *of obtaining the algorithm a as a result of training and the contribution* $Q_\mu(a, A)$ *the algorithm a makes into the overtraining probability preserve for an arbitrary permutation* $\pi \in S_L$ *simultaneously applied to the set A and the algorithm a*

$$P_\mu(a, A) = P_\mu(\pi a, \pi A), \tag{12}$$

$$Q_\mu(a, A) = Q_\mu(\pi a, \pi A). \tag{13}$$

Proof. Note that $\mathbf{E}f(X) = \mathbf{E}f(\pi X)$ holds for the arbitrary function $f(X)$ of the decomposition of the sample $X \sqcup \bar{X}$ into the learning and control samples. We also use the property $\delta(\pi a, \pi X) = \delta(a, X)$ that follows from Lemma 1 and the definition of deviation of frequencies of the algorithm's errors. Then,

$$\begin{aligned} Q_\mu(\pi a, \pi A) &= \mathbf{E}\mu(\pi A, X, \pi a)\delta(\pi a, X) \ge \varepsilon \\ &= \mathbf{E}\mu(\pi A, \pi X, \pi a)[\delta(\pi a, \pi X) \ge \varepsilon] \\ &= \mathbf{E}\mu(A, X, a)[\delta(a, X) \ge \varepsilon] = Q_\mu(a, A). \end{aligned}$$

The equality $P_\mu(\pi a, \pi A) = P_\mu(a, A)$ is obtained from the expression $Q_\mu(a, A) = Q_\mu(\pi a, \pi A)$ by putting $\varepsilon = -1$.

Corollary 1. The overtraining probability preserves when an arbitrary permutation $\pi \in S_L$ is applied to the set of algorithms

$$Q_\mu(A) = Q_\mu(\pi A). \tag{14}$$

Proof.

$$Q_\mu(\pi A) = \sum_{a \in \pi A} Q_\mu(a, \pi A)$$

$$= \sum_{a \in A} Q_\mu(\pi a, \pi A) = \sum_{a \in A} Q_\mu(a, A) = Q_\mu(A).$$

The latter proposition looks very natural since the order of objects in the sample does not matter in most problems of learning from precedents.

2.2. Group of Symmetry of the Set of Algorithms

We recall that we defined the action of the group S_L on the set of all possible sets of algorithms $2^{\mathbb{A}}$ above.

Definition 3. The stationary subgroup Sym(A) of the set of algorithms $A \in 2^{\mathbb{A}}$ is called its *group of symmetry*

$$\mathrm{Sym}(A) = \{\pi \in S_L : \pi A = A\}.$$

Example. We consider the set of algorithms given by the matrix of errors

$$\begin{array}{c|ccccc|} & a_1 & a_2 & a_3 & a_4 & a_5 \\ x_1 & 1 & 1 & 1 & 0 & 0 \\ x_2 & 0 & 1 & 1 & 1 & 0 \\ x_3 & 0 & 0 & 1 & 1 & 1 \\ x_4 & 1 & 0 & 0 & 1 & 1 \\ x_5 & 1 & 1 & 0 & 0 & 1 \end{array}.$$

The rows of the matrix correspond to the objects of the general sample $\mathbb{X}$; the columns, to the algorithms $a \in A$. The group of symmetry of this set is a dihedral group: $\mathrm{Sym}(A) \cong S_2 \ltimes \mathbb{Z}/5\mathbb{Z}$. The circular permutation $\pi_{\circlearrowright} = (x_1, x_2, x_3, x_4, x_5)$ and a pair of transpositions $\pi_{\leftrightarrow} = (x_2, x_5)(x_3, x_4)$ are the group-forming elements.

It is significant that the group of symmetry Sym(A) *acts* upon the set of algorithms A. Indeed, each element of the group of symmetries $\pi \in \mathrm{Sym}(A)$ permutates the algorithms a only *inside* the set A. Hence, for any $a \in A$ and any $\pi \in \mathrm{Sym}(A)$, $\pi a \in A$ holds. Therefore, for the group Sym(A), unlike the entire S_L, the action on the set A is given naturally.

The subset G is called the *orbit* of the element m of the set M, on which the group $Gm = \{gm: g \in G\} \subseteq M$ acts. Either the orbits of two elements m_1 and m_2 do not overlap or they coincide. This allows us to speak of the decomposition of the set M into the nonoverlapping orbits $M = Gm_1 \sqcup \ldots \sqcup Gm_k$.

In what follows, we consider the orbits of action of the group of symmetry Sym(A) on the set of algorithms. We denote all the orbits of the set of algorithms A by $\Omega(A)$. We denote the representative of the orbit $\omega \in \Omega(A)$ by $a_\omega \in A$.

Points of one orbit are called equivalent in group theory. Since, in [1], algorithms with equal vectors of errors on the general sample $\mathbb{X}$ are also called *equivalent*, we call different representatives of the same orbit *identical algorithms*.

Lemma 3. *Identical algorithms have the same number of errors on the complete sample.*

The proof of the lemma automatically follows from Lemma 1

$$n(a, \mathbb{X}) = n(\pi a, \pi X) = n(\pi a, X).$$

By the definition given above, the algorithm $a \equiv (a(x_i))_{i=1}^{L}$ is the vector, and, hence, it depends on the numeration of the objects of the sample. However, neither the group of symmetries Sym(A) nor the decomposition into classes of identical algorithms $\Omega(A)$ depend on this numeration.

Lemma 4. *For any set of algorithms $A \in 2^{\mathbb{A}}$ and any permutation $\pi \in S_L$, the groups* Sym(A) *and* Sym(πA) *are conjugate*: $\mathrm{Sym}(\pi A) = \pi \circ \mathrm{Sym}(A) \circ \pi^{-1}$.

This lemma is equivalent to the known proposition of the group theory; i.e., the stationary subgroups of points belonging to one orbit can be obtained from each other by conjugation [7].

Lemma 5. *Let the algorithms a_1 and a_2 be identical in the set of algorithms A. Then, $\forall \pi \in S_L$ the algorithms πa_1 and πa_2 are identical in the set of algorithms πA.*

Let $\gamma \in \mathrm{Sym}(A)$be the permutation such that $a2 = \gamma a_1$. Then, $\pi a_2 = \pi\gamma a_1 = (\pi\gamma\pi^{-1})\pi a_1 = \tilde{\gamma}\,\pi a_1$. We have from Lemma 4 that $\tilde{\gamma} = \pi\gamma\pi^{-1}$ is the element of Sym(πA).

2.3. Theorems on the Equal Contribution of Identical Algorithms into the Overtraining Probability

The theorems we give in this section may be significantly simplified, in some cases, obtaining explicit formulas for the overtraining probability.

Theorem 3. Identical algorithms have the same probability to be implemented as a result of training and make the same contribution to the overtraining probability

$$P_\mu(a, A) = P_\mu(\pi a, A), \qquad (15)$$

$$Q_\mu(a, A) = Q_\mu(\pi a, A), \qquad (16)$$

where $\pi \in \mathrm{Sym}(A)$.

The proof automatically follows from Lemma 2 and the definition of the group of symmetry $P_\mu(\pi a, A) = P_\mu(\pi a, \pi A) = P_\mu(a, A)$, and similarly for $Q_\mu(a, A)$.

Corollary 2. Let the group of symmetry act on the set of algorithms transitively $A = \{\pi a_0, \pi \in \mathrm{Sym}(A)\}$, where $a_0 \in A$ is the arbitrary algorithm of the set A. Then, all algorithms of the set have the same probability to be implemented as a result of training.

Theorem 3 allows us to move from summing over all algorithms of the set to summing over the orbits of the group Sym(A).

Theorem 4. *We can write the overtraining probability $Q_\mu(A)$ for the randomized method of empirical risk minimization as*

$$Q_\mu(A) = \sum_{\omega \in \Omega(A)} |\omega| \mathbf{E} \frac{[a_\omega \in A(X)]}{|A(X)|}[\delta(a_\omega, X) \ge \varepsilon]. \qquad (17)$$

We apply the theorem on the equivalent contribution identical algorithms make into the overtraining probability followed by definitions (9) and (7)

$$Q_\mu(A) = \sum_{a \in A} Q_\mu(a, A) = \sum_{\omega \in \Omega(A)} |\omega| Q_\mu(a_\omega, A)$$

$$= \sum_{\omega \in \Omega(A)} |\omega| \mathbf{E} \frac{[a_\omega \in A(X)]}{|A(X)|}[\delta(a_\omega, X) \ge \varepsilon].$$

Formula (17) is the main tool to derive accurate estimates of the overtraining probability for the randomized method of empirical risk minimization.

3. ACCURATE ESTIMATES OF THE OVERTRAINING PROBABILITY

In this section, we obtain explicit combinatorial formulas for the functional $Q_\mu(\varepsilon, A)$ for some sets of algorithms A possessing the property of symmetry.

3.1. The Complete Layer of Algorithms

The set consisting of all algorithms $a \in \mathbb{A}$ with a fixed number of errors $n(a, \mathbb{X}) = m$ is called the *complete m-layer* of algorithms.

Theorem 5. *For training by the empirical risk minimization method, the overtraining probability for the complete m-layer of algorithms is*

$$Q_\mu(\varepsilon, A) = [\varepsilon k \le m \le L - \varepsilon l]. \tag{18}$$

Proof. In this case, the entire symmetric group S_L is the group of symmetry Sym(A). Hence, the action of the group of symmetry on the set of algorithms is transitive, and the set has only one class of identical algorithms from C_L^m. By theorem 4, we write

$$Q_\mu(\varepsilon, A) = C_L^m \mathbf{E}\frac{[a_0 \in A(X)]}{|A(X)|}[\delta(a_0, X) \ge \varepsilon],$$

where a_0 is an arbitrary algorithm of the family.

The algorithm a_0 will be chosen only if it makes a minimal number of errors in the course of training. We consider two cases.

Case 1. $m \le k$. All errors of a_0 go to the control sample, with overtraining happening if $m \ge \varepsilon k$. This fixes m objects of the control. Hence, the number of summands in the sum over decompositions X is the number of ways to choose $k - m$ objects such that the algorithm a_0 does not make any error. This number is C_{L-m}^{k-m}.

The cardinality of the set of the algorithms $A(X)$ best in training is independent of X and is C_k^m, i.e., the number of ways to place m errors of the algorithms in k positions of the control sample. Thus,

$$Q_\mu(\varepsilon, A) = \frac{C_L^m C_{l-m}^{k-m}}{C_L^l \; C_k^m}[m \ge \varepsilon k] \;\; \text{for} \;\; m \le k.$$

Case 2. $m > k$. The control sample must consist only of objects on which a_0 makes errors. Then, there are $m - k$ errors left in the training process, and the overtraining condition is $1 - \frac{m-k}{l} \ge \varepsilon$. Hence, $m \le L - \varepsilon l$.

The number of decompositions of the sample, for which $a_0 \in A(X)$, is C_m^k, i.e., the number of ways to choose k errors of the algorithm a_0 for the control sample. The cardinality of the set $A(X)$ again does not depend on X and is C_l^{m-k}, i.e., the number of ways to choose $m - k$ errors for the learning sample

$$Q_\mu(\varepsilon, A) = \frac{C_L^m C_l^{m-k}}{C_L^l \; C_m^k}[m \le L - \varepsilon l] \;\; \text{for} \;\; m > k.$$

Writing the identity $C_L^k = \frac{L!}{k!(L-k)!}$ for each combinatorial coefficient, we find that the combinatorial factors are unity in both formulas. Combining the conditions $\varepsilon k \ge m \ge k$ and $k < m \le L = \varepsilon l$, we obtain the theorem's hypothesis.

3.2. Cube of Algorithms

We call the cube of algorithms $\mathbb{A}$ the set that includes all possible Boolean vectors $a \in \{0, 1\}^L$.

Theorem 6. *The overtraining probability for the cube of algorithms is given by the formula*

$$Q_\mu(\varepsilon, \mathbb{A}) = \frac{1}{2^k}\sum_{m = \lceil \varepsilon k \rceil}^{k} C_k^m.$$

Proof. Obviously, in this case, the group of symmetry is the entire S_L. Then, the layers of algorithms with the same number of errors are its orbits. Therefore, by Theorem 4,

$$Q_\mu(\varepsilon, \mathbb{A}) = \sum_{m=0}^{L} C_L^m \mathbf{E}\frac{[a_m \in A(X)]}{|A(X)|}[\delta(a, X) \ge \varepsilon].$$

The algorithm can be chosen as a result of training only if it makes no errors in the course of training. Therefore, all its errors should go to the control sample. Hence, we can limit the summation index to $m \le k$.

Since all errors of the chosen algorithm are in the control sample, the deviation of the frequencies is $\delta(a, X) = \frac{m}{k}$ for any decomposition. Hence, the overtraining happens for $m \ge \lceil \varepsilon k \rceil$.

There are always 2^k algorithms in the set $A(X)$. The algorithms have zero errors in the course of training and all possible vectors of errors on the control sample.

Combining all the established facts, we obtain the formula

$$Q_\mu(\varepsilon, \mathbb{A}) = \sum_{m = \lceil \varepsilon k \rceil}^{k} C_L^m \frac{\mathbf{E}[a_m \in A(X)]}{2^k}.$$

Now, we have to calculate the number of decompositions on which the algorithm a_m is chosen by the learning method. This is the number of ways to choose l objects of the learning sample from L correct answers of the algorithm a_m. Finally, we have

$$Q_\mu(\varepsilon, \mathbb{A}) = \sum_{m = \lceil \varepsilon k \rceil}^{k} C_L^m \frac{C_{L-m}^l}{C_L^l 2^k} = \frac{1}{2^k}\sum_{m = \lceil \varepsilon k \rceil}^{k} C_k^m.$$

3.3. Unimodal Chain

We find the distance between algorithms $\rho(a, a')$ as the Hamming distance between their vectors of errors

$$\rho(a, a') = \sum_{x \in X} |a(x) - a'(x)|.$$

Definition 4. The set of algorithms $\{a_0, ..., a_D\}$ is called a *monotonic chain* if two conditions are met:

(1) the number of errors is monotonic: $n(a_i, \mathbb{X}) = m + i$, $i = 0, ..., D$ for some fixed m,

(2) errors of the previous algorithm are absorbed: $\rho(a_i, a_{i-1})$, $i = 1, ..., D$.

Thus, every subsequent algorithm in the monotonic chain makes errors on the same objects as the previous algorithm and one additional error.

The monotonic chain of algorithms is the simplest model of a one-parametric *connected family of algorithms* implying that the number of errors on the complete sample increases as some parameter continuously moves away from its optimal value.

Definition 5. The set of algorithms $\{a_0, a_1, ..., a_D, a'_1, ..., a'_D\}$ is called a unimodal chain if two conditions are met:

(1) its left $\{a_0, a_1, ..., a_D\}$ and right $\{a_0, a'_1, ..., a'_D\}$ branches are monotonic chains,

(2) the overlapping of the set of errors of algorithms a_D and a'_D is the set of errors of the algorithm a_0.

The unimodal chain is a more realistic model of the one-parametric *connected family* as compared to the monotonic chain. If we have the best algorithm a_0 with the optimal value of some real parameter, the deviation of the value of this parameter both upward and downward makes the number of errors increase.

Theorem 7. *For the unimodal chain with branches of length D, the overtraining probability of the randomized method of empirical risk minimization is*

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D}\sum_{t_1=h}^{D}\sum_{t_2=0}^{D} \frac{|\omega_h|}{1 + t_1 + t_2} \frac{C_{L'}^{l'}}{C_L^l} H_{L'}^{l', m}(s(\varepsilon)), \quad (19)$$

where $L' = L - t_1 - t_2 - F$, $F = [t_1 \neq D] + [t_2 \neq D]$, $l' = l - F$, $s(\varepsilon) = \left\lfloor \frac{l}{L}(m + h - \varepsilon k) \right\rfloor$, $|\omega_h| = 1$ *for* $h = 0$ *and* $|\omega_h| = 2$ *for* $h \geq 1$; $H_{L'}^{l', m}(z) = \frac{1}{C_{L'}^{l'}} \sum_{s=0}^{\lfloor z \rfloor} C_m^s C_{L'-m}^{l'-s}$ *is the function of hypergeometric distribution* [4].

Proof. We numerate the objects of the general sample $\mathbb{X}$ as shown in the table

$$\begin{array}{c|ccccccccc|}
 & a_0 & a_1 & a_2 & \ldots & a_D & a'_1 & a'_2 & \ldots & a'_D \\
x_1 & 0 & 1 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \\
x_2 & 0 & 0 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \\
 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
x_D & 0 & 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 \\
\hline
x'_1 & 0 & 0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 \\
x'_2 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 & \ldots & 1 \\
 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
x'_D & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 1
\end{array}.$$

Permutating the objects of the sample ($x_1 \longleftrightarrow x'_1$, ..., $x_D \longleftrightarrow x'_D$), we can rearrange the left and right branches. Therefore, identical in the unimodal chain are the pairs of algorithms with the same number of errors on the complete sample.

By Theorem 4, we can write the overtraining probability as

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D}\sum_{t_2=0}^{D} \frac{1}{C_L^l}$$

$$\times \sum_{X \in N(t_1, t_2)} \frac{1}{|A(X)|}[\delta(a_h, X) \geq \varepsilon].$$

Here, the index h denotes the number of the class of identical algorithms (so that all algorithms of the class ω_h have $m + h$ errors); $|\omega_0| = 1$, and $|\omega_h| = 2$ for $h \geq 1$. For the sake of definiteness, we take the representative a_h of the class ω_h from the left branch of the chain.

The indices t_1 and t_2 parameterize the content of the set $A(X)$. For an arbitrary decomposition $X \in [\mathbb{X}]^l$, we give t_1 as the maximal number for which all objects $x_1, x_2, ..., x_t$ are in the control sample and x_{t_1+1}, if it exists, is in the learning sample. The index t_2 is given similarly for the objects of the right branch. The set $N(t_1, t_2) \subset [\mathbb{X}]^l$ is the set of all decompositions with the parameters t_1 and t_2.

It follows from the definition of t_1 and t_2 that $|A(X)| = \frac{1}{1 + t_1 + t_2}$. Within the summation, the indices t_1 and t_2 take various sets of values since we consider only those decompositions for which the representative a_h chosen from the left branch lies in $A(X)$.

We denote $F = [t_1 \neq D] + [t_2 \neq D]$, $L' = L - t_1 - t_2 - F$, $l' = l - F$. The parameter F allows us to take into account the contribution of the last algorithms a_D and a'_D of the chain.

We calculate the cardinality of the subset of those decompositions from $N(t_1, t_2)$ on which the algorithm a_h is overtrained. Let $s_0(\varepsilon)$ be the maximal number of errors in the course of training for which we have overtraining. By the definition of deviation of frequencies, we find $s_0(\varepsilon) = \left\lfloor \frac{l}{L}(m + h - \varepsilon k) \right\rfloor$. We need to choose l' objects from L' for training so that there are no more than $s_0(\varepsilon)$ errors of m free errors of the algorithm a_h in the training. The number of ways is $\sum_{s=0}^{s_0(\varepsilon)} C_m^s C_{L'-m}^{l'-s}$.

Combining all results, we obtain the final formula

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \frac{1}{1 + t_1 + t_2} \frac{C_{L'}^{l'}}{C_L^l} H_{L'}^{l', m}(s_0(\varepsilon)).$$

3.4. Bundle of Monotonic Chains

A bundle of p monotonic chains is the set of algorithms obtained by combining p monotonic chains of the same length with the common first algorithm. As for the unimodal chain, we assume that the sets of objects on which the algorithm makes errors do not overlap.

The group of symmetry of the bundle of p monotonic chains is the symmetrical group S_p that acts upon the branches of the bundle by various permutations. Thus, the classes of identical algorithms are the subsets of algorithms with the same number of errors on the complete sample called *layers* [4].

The following theorem gives the explicit formula for the overtraining probability for the bundle of p monotonic chains. We introduce a *combinatorial coefficient* $R_{D,p}^h(S, F)$ that depends on the parameters S and F, on the number of monotonic chains p, and on their length D, as well as on h, which is the minimal value of the parameter S. The coefficient $R_{D,p}^h(S, F)$ is the number of ways to represent the value S as the sum of p nonnegative summands, $S = t_1 + \ldots + t_p$, each of which does not exceed D. F summands exactly should not equal D, with an additional restriction $t_1 \ge h$ imposed on the first summand.

Theorem 8. *Let the bundle of p monotonic chains have the best algorithm making m errors on the complete sample, and the length of each branch is D, leaving the best algorithm aside. Then, for training performed by the randomized method, we can write the overtraining probability as*

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D} \sum_{S=h}^{pD} \sum_{F=0}^{p} \frac{|\omega_h| R_{D,p}^h(S, F)}{1 + S} \times \frac{C_{L'}^{l'}}{C_L^l} H_{L'}^{l', m}(s(\varepsilon)), \qquad (20)$$

where $L' = L - S - F$, $l = l - F$, $s(\varepsilon) = \left\lfloor \frac{l}{L}(m + h - \varepsilon k) \right\rfloor$; $|\omega_h| = 1$ *for* $h = 0$ *and* $|\omega_h| = p$ *for* $h \ge 1$; $H_{L'}^{l', m}(s)$ *is the function of hypergeometric distribution* [4].

Proof. Generalizing the reasoning given for the unimodal chain in a natural way, we obtain the formula

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D} |\omega_h| \times \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \cdots \sum_{t_p=0}^{D} \frac{1}{1 + t_1 + t_2 + \ldots + t_p} \frac{C_{L'}^{l'}}{C_L^l} H_{L'}^{l', m}(s(\varepsilon)),$$

where $L' = \mathrm{L} - \sum_{i=1}^{p} t_i - \sum_{i=1}^{p} [t_i \ne D]$, $l' = l - \sum_{i=1}^{p} [t_i \ne D]$, $s_0(\varepsilon) = \left\lfloor \frac{l}{L}(m + h - \varepsilon k) \right\rfloor$.

We use the additional designation $S = \sum_{i=1}^{p} t_i$, $F = \sum_{i=1}^{p} [t_i \ne D]$ to simplify the writing. The parameter S gives the cardinality of the set $A(X)$.

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D} |\omega_h| \times \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \cdots \sum_{t_p=0}^{D} \frac{1}{1 + S} \frac{C_{L'}^{l'}}{C_L^l} H_{L'}^{l', m}(s_0(\varepsilon)),$$

where $L' = L - S - F$, $l' = l - F$, $s_0(\varepsilon) = \left\lfloor \frac{l}{L}(m + h - \varepsilon k) \right\rfloor$.

Now, we can pass from summing over the parameters t_i to summing over the set of all possible values S and F

$$Q_\mu(\varepsilon, A) = \sum_{h=0}^{D} |\omega_h| \sum_{S=h}^{pD} \sum_{F=0}^{p} \frac{R_{D,p}^h(S, F)}{1 + S} \frac{C_{L'}^{l'}}{C_L^l} H_{L'}^{l', m}(s_0(\varepsilon)),$$

where $R_{D,p}^h(S, F)$ is the combinatorial coefficient defined above.

The bundle of $2p$ monotonic chains is a model of the p-parametric family of algorithms, in which one can change any of p parameters for other parameters fixed and cannot change several parameters at a time. This family can be treated as the generalization of three special cases considered in [3], i.e., the monotonic chain ($p = 1$), the unimodal chain ($p = 2$), and the unit neighborhood of the best algorithm ($D = 1$).

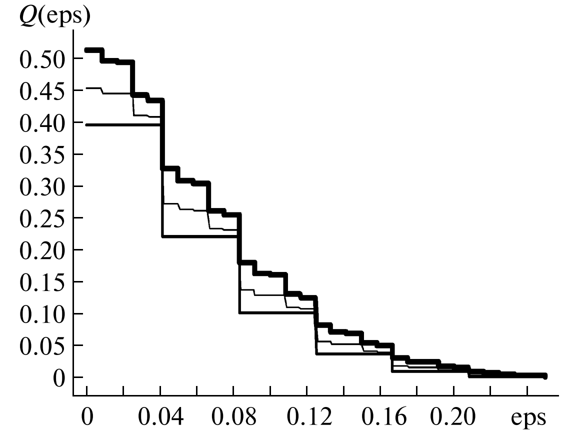

Fig. 1. $Q_\mu(\varepsilon, A)$ depending on ε for the monotonic chain for $L = 100$, $l = 60$, $D = 40$, $m = 20$.

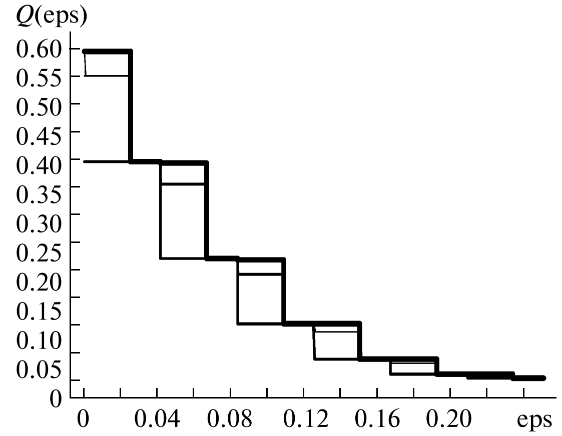

Fig. 2. $Q_\mu(\varepsilon, A)$ depending on ε for the unit neighborhood for $L = 100$, $l = 60$, $p = 10$, $m = 20$.

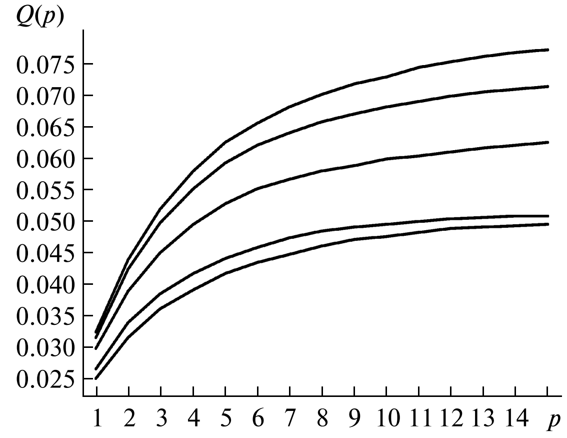

Fig. 3. $Q_\mu(\varepsilon, A)$ depending on p for the bundle of monotonic chains for $L = 300$, $l = 150$, $D = 1, 2, 3, 5, 10$, $\varepsilon = 0.05$.

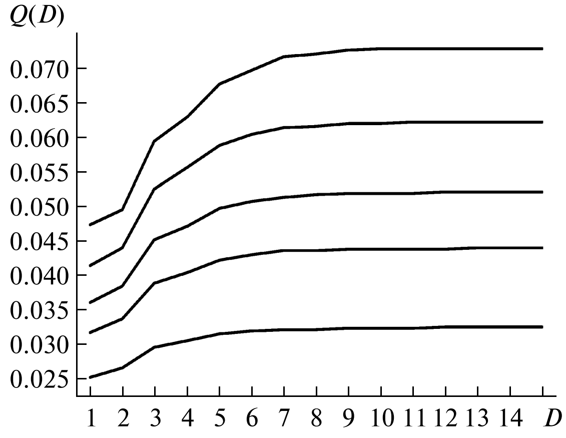

Fig. 4. $Q_\mu(\varepsilon, A)$ depending on D for the bundle of $p = 1, 2, 3, 5, 10$ monotonic chains for $L = 300$, $l = 150$, $m = 15$, $\varepsilon = 0.05$.

The formula for the overtraining probability for the unimodal chain was obtained in Theorem 7. To obtain explicit formulas for the two other families, it is sufficient to find the explicit expression for the combinatorial coefficient $R_{D,p}^h(S, F)$.

Corollary 3. For the monotonic chain of length $D + 1$, the overtraining probability is

$$Q_\mu(\varepsilon, A) = \frac{1}{C_L^l}\sum_{h=0}^{D}\sum_{S=h}^{D}\frac{1}{1+S}H_{L'}^{l',m}(s(\varepsilon)), \qquad (21)$$

where $L' = L - S - [S \neq D]$, $l' = l - [S \neq D]$.

Corollary 4. For the unit neighborhood of $p + 1$ algorithms, the overtraining probability is

$$Q_\mu(\varepsilon, A) = \frac{1}{C_L^l}\sum_{h=0}^{D}\sum_{S=h}^{D}\frac{|\omega_h| C_{p-h}^{S-h}}{1+S}H_{L'}^{l',m}(s(\varepsilon)), \qquad (22)$$

where $L' = l - p$, $l' = l + S - p$.

3.5. Numerical Experiment

Figures 1 and 2 give the results of numerical experiments that compared overtraining probabilities for different options of empirical risk minimization. Each graph has four curves, with the upper (bold) one corresponding to pessimistic empirical risk minimization [3, 4] and the lower one, to optimistic empirical risk minimization. Two almost coinciding curves that lie between them stand for the randomized empirical risk minimization. One of them is calculated using the proved formulas, and the second one is constructed by the Monte Carlo method using 10^5 random decompositions, given the equiprobable choice of the best algorithm in the case of uncertainty. The differences of these two curves lie within the error of the Monte Carlo method.

Figures 3 and 4 give the overtraining probability depending on the number of branches p in the bundle and their length D. The graphs are drawn for the randomized empirical risk minimization method. Figure 4 shows that when the length of chains D increases, the overtraining probability almost stops growing as early as for $D = 7$. This is due to the *localization effect* [4], i.e., only algorithms from lower layers have a substantially nonzero probability to be chosen by the empirical risk minimization method. Adding algorithms that are "too bad" does not increase the overtraining probability. Figure 3 shows that the overtraining probability continues growing as the number of chains p in the bundle grows. However, the *connectivity effect* makes the growth rate sublinear with respect to p; i.e., all algorithms are at a Hamming distance not greater than D from the best algorithm.

4. CONCLUSIONS

With the symmetry property of families of algorithms, we can obtain computationally efficient formulas for the overtraining probability. For the monotonic chain, the unimodal chain, and the unit neighborhood, we obtained formulas such as the corollary of one theorem, while similar estimates used to be proved independently and under a nonnatural assumption regarding a priori arrangement of algorithms in the family [3]. The proposed approach allows obtaining estimates for the family with an exponentially growing number of algorithms (the complete cube, the Boolean cube).

ACKNOWLEDGMENTS

This study was supported by the Russian Foundation for Basic Research (project no. 08-07-00422) and the program of Department of Mathematics of the Russian Academy of Sciences "Algebraic and Combinatorial Methods of Mathematical Cybernetics and New-Generation Information Systems."

REFERENCES

1. V. N. Vapnik and A. Ya. Chervonenko, *Pattern Recognition Theory* (Moscow, Nauka, 1974) [in Russian].
2. V. Vapnik, *Statistical Learning Theory* (New York, Wiley, 1998).
3. K. V. Vorontsov, "Accurate Estimates Of Overtraining Probability," Doklady RAN, **429** (1), 15–18 (2009).
4. K. V. Vorontsov, "Combinatorial Approach to Overtraining," All-Russian Conference on Mathematical Methods of Pattern Recognition-14 (Moscow, MAKS Press, 2009), pp. 18–21.
5. P. V. Botov, "Accurate Estimates of Overtraining Probability for Monotonic and Unimodal Families of Algorithms," All-Russian Conference on Mathematical Methods of Pattern Recognition-14 (Moscow, MAKS Press, 2009), pp. 7–10.
6. A. I. Frei, "Accurate Estimates of Overtraining Probability for Symmetric Families of Algorithms," All-Russian Conference on Mathematical Methods of Pattern Recognition-14 (Moscow, MAKS Press, 2009), pp. 66–69.
7. E. B. Vinberg, *Algebra* (Moscow, Factorial Press, 2001) [in Russian].

Alexander I. Frei was born in 1987. He graduated from Moscow Institute of Physics and Technology (Department of Control and Applied Mathematics) in 2010. He received his master degree (Applied Physics and Mathematics) in 2010. He was awarded the second prize for young scientists at the 14th Conference on Mathematical Methods and Pattern Recognition (September 2009). His scientific interests include statistical learning theory, machine learning, data mining, probability theory, and combinatorics. He is the author of one publication.